Research report 122

PLANAR ACYCLIC COMPUTATION

by

W.F.McColl[1]

and

Michael S. Paterson[2]

(RR 122)

Abstract

This paper considers the following problem: given a specification consisting of a set of variables X, a multiset of functions F on those variables, and a cyclic ordering on $X \cup F$, determine whether or not there exists a planar acyclic circuit which realises the specification. An algorithm is given which produces such a circuit whenever one exists. In proving that our algorithm meets this requirement we provide a simple mathematical characterisation of those specifications which are realisable.

Department of Computer Science
University of Warwick
Coventry, CV4 7AL, England

May 1988

[1]Programming Research Group, Oxford University, 11 Keble Road, Oxford, UK.
[2]Department of Computer Science, University of Warwick, Coventry, UK.
[This author was supported by a Senior Fellowship of the SERC.]

Planar Acyclic Computation

W.F.McColl
Programming Research Group
Oxford University
11 Keble Road
Oxford OX1 3QD
England

M.S.Paterson
Department of Computer Science
University of Warwick
Coventry CV4 7AL
England

April 1988

Abstract

This paper considers the following problem : given a specification consisting of a set of variables X, a multiset of functions F on those variables, and a cyclic ordering on $X \cup F$, determine whether or not there exists a planar acyclic circuit which realises the specification. An algorithm is given which produces such a circuit whenever one exists. In proving that our algorithm meets this requirement we provide a simple mathematical characterisation of those specifications which are realisable.

Keywords : arithmetic functions, Boolean functions, circuit, complexity, computability, crossover, embedding, graph, planar circuit, VLSI.

1 Introduction

Every Boolean function of n arguments can be computed by an acyclic circuit constructed from input nodes corresponding to the arguments, and two-input gates corresponding to binary Boolean functions. Every polynomial over a field F can be computed by an acyclic circuit constructed from input nodes corresponding to the arguments, constant nodes corresponding to the elements of F, and two-input gates corresponding to the binary operations $\{+,-,*,/\}$. Such universality properties have resulted in the acceptance of acyclic circuits as a fundamental model of computation for arithmetic and Boolean functions. In recent years there have been a number of major advances in our understanding of the circuit complexity of functions. For an account of the circuit complexity of Boolean functions, see [We]. Results on the circuit complexity of arithmetic functions can be found in [BM,vzG2].

Circuit complexity has traditionally been measured in terms of circuit size, i.e., the number of gates. However, recent technological developments have resulted in a much greater emphasis being placed on the design of highly parallel circuits which minimise depth (rather than size), and on the design of circuits which can be directly fabricated as VLSI devices. The circuit depth of functions is discussed in [BM,vzG1,MRK,Re,We]. Results on the VLSI complexity of functions can be found in [L,U,We]. Although most of the work on VLSI complexity has assumed a technology which is essentially two-dimensional, with perhaps a fixed number of layers, there has been some research on the problem of embedding circuits in three-dimensional space [GH,LR,P,Ro], and also on the problem of embedding circuits in "books" [Y1,Y2].

In this paper we consider the computation of functions by acyclic circuits embedded in the plane without crossing edges. The planar circuit model of computation was studied by Lipton and Tarjan [LT] who proved a number of interesting and important lower bounds on the planar circuit size of Boolean functions using their separator theorem. In [S1,S2], Savage showed that planar circuit size was closely related to the AT^2 complexity measure used for VLSI circuits [U,We], and, in doing so, simplified and unified a number of earlier lower bounds on VLSI complexity.

Definition

A *planar crossover* is a planar acyclic circuit whose ordered inputs are $\langle x, y\rangle$ and whose ordered outputs are $\langle y, x\rangle$.

Example 1.1 *Figure 1 shows a planar Boolean crossover constructed from gates corresponding to the function* $\oplus(p,q) = (p+q)\,modulo\,2$.

The existence of planar crossovers [LT,McC1,S1] is often taken as sufficient justification for the statement that the planar circuit model is universal. However, Aggarwal [A] made the interesting observation that the conversion of a general acyclic

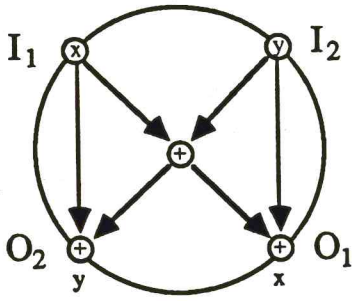

Figure 1

circuit into a planar circuit by the substitution of planar crossovers for crossing edges might introduce a cycle into the graph. For the kinds of planar circuit considered in [LT,S1] where the inputs and outputs are segregated from each other a on the boundary, this problem can easily be resolved by first topologically ordering the gates.

In this paper we consider further the problem raised initially by Aggarwal. We investigate circuits in which the inputs and outputs are interleaved in some way on the boundary of the circuit. Our results show that for many such circuits it is not possible to replace crossing edges by planar crossovers without introducing a cycle. The results are given in terms of an algorithm which, given an input/output specification consisting of

(i) a set of variables X,

(ii) a multiset of functions F on those variables,

(iii) a cyclic ordering on $X \cup F$,

will produce a planar acyclic circuit which realises the specification whenever one exists. In proving that our algorithm meets this requirement we provide a simple mathematical characterisation of those specifications which are realisable. The results established apply to both arithmetic and Boolean circuits.

2 Definitions and Preliminary Results

Definitions

Let $X_n = \langle x_1, x_2, \ldots, x_n \rangle$ be a set of n formal arguments and D be a set of values. $D_n = \{f(X_n) : D^n \rightarrow D\}$ will be used to denote the set of n-argument functions over the domain D.

Let π be a finite region of the plane bounded by a simple closed curve γ. An *I/O specification* consists of a set $I = \{I_1, I_2, \ldots, I_n\}$ of input ports on γ and a disjoint set $O = \{O_1, O_2, \ldots, O_m\}$ of output ports on γ, where each I_j is associated with the argument x_j and each O_j is associated with some $f_j \in D_n$. Note that the cyclic

ordering of the ports on γ is part of the I/O specification. We shall see later that it is a very important part.

A *circuit over the basis* Ω, $\Omega \subseteq D_2$, is a directed acyclic graph where (i) nodes have either in-degree 2 (*gates*) or in-degree 0 (*input nodes*), (ii) each gate has an ordering on its two inputs and corresponds to some function in Ω, and (iii) all output nodes are required to have out-degree 0.

$D_n^{(\Omega)}$ is the set of functions in D_n which can be realised by circuits over the basis Ω.

A *circuit layout for* an I/O specification is a circuit embedded in π as a planar graph with the following properties :

(i) to each I_j, $1 \leq j \leq n$, is mapped a unique input node corresponding to x_j.

(ii) to each O_j, $1 \leq j \leq m$, is mapped an output node computing f_j.

A planar crossover is then a circuit layout for the specification with $n = m = 2$ given by $f_1 = x_1$, $f_2 = x_2$, and the cyclic order I_1, I_2, O_1, O_2.

In [McC1], the set of Boolean bases which permit the realisation of a planar crossover were characterised. For example, there is a planar crossover for the complete basis $\{NAND\}$ and for the incomplete basis $\{\wedge, \rightarrow, \leftarrow\}$. On the other hand there is no planar crossover for the monotone basis $\{\wedge, \vee\}$. The latter result shows that although every (single-output) monotone Boolean function f can be realised by a circuit over the basis $\{\wedge, \vee\}$, it is not necessarily the case that f can be realised by a planar monotone circuit. In [McC2], examples are given of single-output monotone Boolean functions which cannot be realised by any planar monotone circuit. Planar arithmetic crossovers can easily be constructed for many arithmetic bases by using inverse operations, e.g., addition/subtraction or multiplication/division.

An I/O specification is *k-directional* if and only if the boundary γ can be partitioned into $2k$ segments with the cyclic order $\langle \gamma_{I_1}, \gamma_{O_1}, \gamma_{I_2}, \gamma_{O_2}, \ldots, \gamma_{I_k}, \gamma_{O_k} \rangle$ such that every input port is in some γ_{I_j} and every output port is in some γ_{O_j}.

Theorem 2.1 *A specification S has a circuit layout over Ω if the following three conditions hold*

(i) $f_1, f_2, \ldots, f_m \in D_n^{(\Omega)}$,
(ii) there is a planar crossover for Ω,
(iii) S *is 1-directional or 2-directional*

Proof. We first prove the result for the 1-directional case. It is straightforward to construct an acyclic (not necessarily planar) circuit over Ω which computes the set of functions $f_1, f_2, \ldots, f_m$. Let $\langle g_1, g_2, \ldots, g_p \rangle$ be a total ordering of the non-output gates which is consistent with the partial ordering given by the circuit. Let the given cyclic (clockwise) ordering of the ports in the I/O specification be $\langle I_{j_1}, \ldots, I_{j_n}, O_{k_m}, \ldots, O_{k_1} \rangle$. Locate I_{j_t} at $(0, t)$ and O_{k_t} at $(p+1, t)$. Locate each g_t

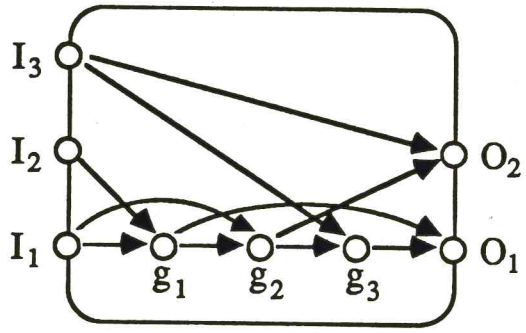

Figure 2

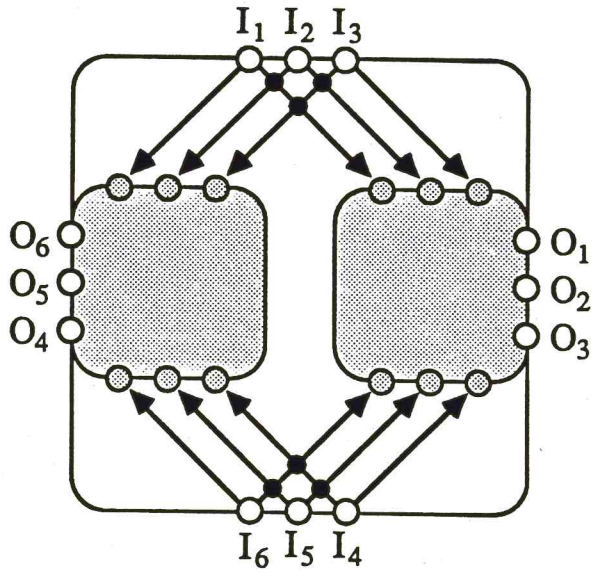

Figure 3

at the point $(t, 1)$ and draw each edge as a path increasing monotonically in x as shown in Fig. 2, arranging the edges so that at most two cross at any point.

Replace each such crossing by an instance of the planar acyclic crossover circuit. This yields an acyclic planar layout and establishes the result for the 1-directional case.

Given a 2-directional I/O specification, use planar crossovers to produce two copies of each block of input ports as illustrated in Fig.3. Now use the construction described above to produce planar acyclic circuits for the resulting pair of 1-directional specifications. □

Definitions

The *dependency relation* $R_S \subseteq I \times O$ for an I/O specification S is defined by $R_S = \{\langle j, k\rangle | \exists a, b \in D, f_{k|x_j=a} \neq f_{k|x_j=b}\}$, i.e., $\langle j, k\rangle \in R_S$ if and only if f_k depends on x_j.

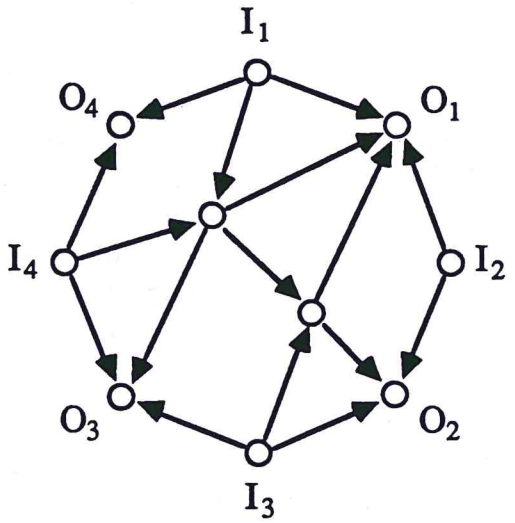

Figure 4

A *diagram* is a disjoint pair I, O of finite sets together with a cyclic ordering on $I \cup O$ and a relation in $I \times O$. In the next section we show that the geometric information necessary to determine whether or not a specification has a circuit layout is captured by its diagram.

3 Main Construction

Definitions

A *wiring layout for* a diagram is a directed acyclic graph embedded in a simple closed region of the plane, with the following properties. Each input in I (output in O) has a corresponding node with in-degree (out-degree) 0 on the boundary of the region; all other nodes are in the interior. The cyclic arrangement of edges at interior nodes is such that the incoming and outgoing edges do not interleave, i.e., the incoming and outgoing edges lie in disjoint sectors in the neighbourhood of each node. For each I/O pair in the diagram there is a directed path between the corresponding nodes in the wiring layout.

Example 3.1 *Figure 4 shows a wiring layout for the diagram of the I/O specification for* $m = n = 4$ *given by the cyclic order of ports* $\langle I_1, O_1, I_2, O_2, I_3, O_3, I_4, O_4 \rangle$ *and the function definitions* $f_1 = f_2 = x_1 \wedge x_2 \wedge x_3 \wedge x_4, f_3 = x_1 \vee x_3 \vee x_4, f_4 = x_1 \wedge x_4$.

Lemma 3.1 *A specification* S *has a circuit layout over* Ω *if the following three conditions hold*

(i) $f_1, f_2, \ldots, f_m \in D_n^{(\Omega)}$,
(ii) there is a planar crossover for Ω,

(iii) there exists a wiring layout for the diagram of S.

Proof. Take a wiring layout for the diagram and choose a simple path following edges of the layout to correspond to each pair in the dependency relation R_S. Replace each edge, which is traversed by k paths say, by k parallel edges and replace each output node O_j by the appropriate acyclic circuit for f_j guaranteed by Theorem 2.1. At an interior node it is necessary to interconnect incoming and outgoing edges to continue the paths. Since incoming and outgoing edges do not interleave, this task corresponds to a 1-directional specification. A local circuit layout at each interior point is therefore also assured by Theorem 2.1, and consists merely of a rearrangement of inputs and outputs using planar crossovers. □

In the remainder of the paper we show that if there exists a wiring layout for some diagram then it can be produced by following a sequence of steps each of which corresponds to a simplification of the diagram.

Let D_0 be the diagram of a specification S_0. Our construction is described in terms of three operations *Extract*, *Prune* and *Coalesce*. Each of these takes a diagram D_i and produces a new diagram D_{i+1} together with a partial wiring layout which relates the two diagrams.

Extract

This operation can be applied whenever there is an adjacent pair of related ports in D_i. If I_j and O_k are adjacent and related, then we can extract this relationship as follows. D_{i+1} is obtained from D_i by eliminating the pair $\langle j, k \rangle$. A wiring layout W_i for D_i can then be obtained from a wiring layout W_{i+1} for D_{i+1} by adding an edge from I_j to O_k running so close to the perimeter that it does not intersect any edge of the layout W_{i+1}.

Prune

This operation can be applied to any port P with no related ports. D_{i+1} is just D_i with that port removed, and a wiring layout for D_i is obtained merely by placing a new node for P just outside the perimeter of D_{i+1} between the neighbours of P and extending the perimeter appropriately.

Coalesce

This operation is applicable where there is a pair of adjacent input (output) ports. We will suppose that I_j and I_k are adjacent. (The case for output ports is analogous.) In D_{i+1} both ports are removed and a new port I' substituted. I' appears in the cyclic order in the position of the ports it replaces, and is related to all output ports which were related to either I_j or I_k. A wiring layout for D_i can be produced by adding new nodes for I_j and I_k, appropriately ordered, just outside the node corresponding to I' in a layout for D_{i+1}, and drawing directed edges to it from each of the new nodes. (See Fig. 5.)

These three operations can be applied in any order and the construction succeeds if a sequence of extraction, pruning and coalescing steps results in the empty dia-

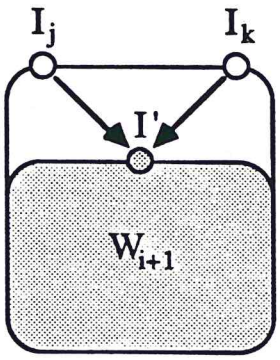

Figure 5

gram. It fails if we reach a stage where we have a nonempty diagram and no further operation can be performed. In the next section we show that if the construction fails then a wiring layout is impossible.

4 Main Theorem

Definitions

A *reduction* is a sequence of extractions, prunings and coalescings. An *irreducible diagram* is a nonempty diagram in which no reductions are possible, so that in particular the input and output ports must alternate. A *k-necklace* is a set of k, $k > 2$, arcs $\langle I_1, O_1\rangle, \langle I_2, O_2\rangle, \ldots, \langle I_k, O_k\rangle$ such that $\langle I_1, O_n, I_2, O_1, I_3, \ldots, O_{n-2}, I_n, O_{n-1}\rangle$ are in cyclic order.

Let D be an irreducible diagram. The *span* of an arc in D is j if there are exactly j inputs (or equivalently outputs) between the ends of the arc in the shorter direction. For arcs p, q, *p releases q* if, when p is removed and any possible prunings and coalescings are made, q is extractable.

The next result follows from the definitions.

Lemma 4.1 *If p releases q then q has span 1 and one of the two ports in the span of q is only incident with p.* □

Theorem 4.1 *Every irreducible diagram contains a necklace.*

Proof. Assume that D contains no necklace and has the smallest number of arcs for such an irreducible diagram. It follows that D has more than three arcs. Every arc p in D must release at least one other arc, for otherwise the result of removing p and reducing the remaining diagram would be a smaller necklace-free irreducible diagram. Suppose in D that r releases p and p releases q, where possibly $r = q$. Without loss of generality we have one of the following structures (Figs. 6 and 7), where $I_1, O_1, I_2, O_2, I_3, O_3$ are successive ports.

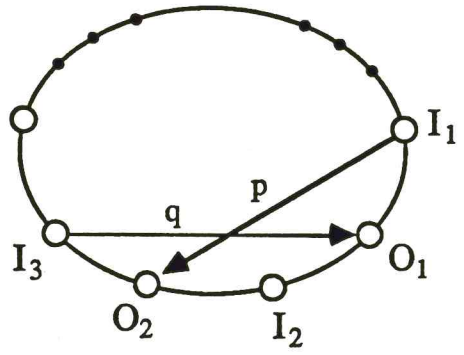

Figure 6

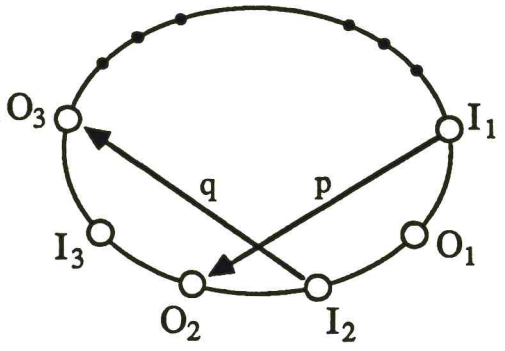

Figure 7

Figure 6 is impossible since there must be some non-extractable arc s leaving I_2, and so $\{p, q, s\}$ would be a 3-necklace. In Figure 7, suppose we replace arcs p and q by a single arc pq from I_1 to O_3. By Lemma 4.1, port O_2 no longer has an incident arc, so we can prune O_2 and coalesce ports I_2 and I_3 to a new port I'. Denote the resulting diagram by D'. (See Figure 8.)

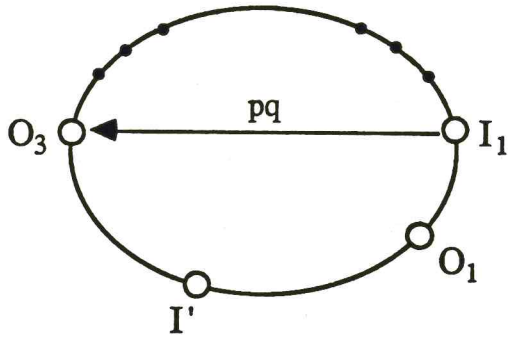

Figure 8

Now if D' has a k-necklace N then either pq is not an arc of N and N gives a k-necklace in D, or else pq is an arc of N. In the latter case, if the necklace arc out of I' came out of I_2 in D then the replacement of pq in D' by the arc p in D yields a k-necklace in D, while if the necklace arc was from I_3 then putting the original pair p and q in place of pq gives a $(k+1)$-necklace.

On the other hand, if D' has no necklace then, by the hypothesis on D, it must be reducible. The only possibilities for a reduction that present themselves in D' are the extraction of an arc (I', O_1) arising from an arc $s = (I_3, O_1)$ in D, or the extraction of arc pq. If pq were extractable then I_1 and O_3 would be adjacent ports and the only possible destination for the arc from I_3 in Figure 7 would be O_1. However, in either case the arc (I_3, O_1) together with p and q would give a 3-necklace in D. □

Theorem 4.2 *No planar circuit layout is possible for an I/O specification whose diagram contains a necklace.*

Proof. Suppose there exists a planar acyclic circuit for a specification whose diagram contains an l-necklace, $\langle I_1, O_1\rangle, \langle I_2, O_2\rangle, \ldots, \langle I_l, O_l\rangle$. Without loss of generality, let the variables corresponding to the necklace arcs be $\langle x_1, x_2, \ldots, x_l\rangle$ respectively.

First we note that there must be a gate in the circuit which depends on two of the necklace variables. If there is no such gate then we have a path from I_1 to O_1 on which each gate is dependent on x_1 but independent of x_2. We also have a path from I_2 to O_2 on which all gates are dependent on x_2 but independent of x_1. Since these paths must cross we have a contradiction.

Therefore, there must be a gate which depends on some pair of necklace variables. Moreover, by acyclicity, there must be such a gate which is not preceded in the partial order by any other such gate. Let the pair of necklace variables for such an earliest gate be x_j, x_k. Since $l > 2$ we may suppose, without loss of generality, that $1 < j < k \leq l$. The ports $I_{j-1}, I_j, O_{j-1}, I_k$ occur in cyclic order. Now we have a contradiction since there is a path from I_{j-1} to O_{j-1} on which each gate depends on x_{j-1}, and an undirected path between I_j and I_k on which each gate depends on no necklace variable other than x_j or x_k. □

Corollary 4.1 *The "triple crossover" given by the I/O specification with* $m = n = 3$, *cyclic order of ports* $\langle I_1, O_3, I_2, O_1, I_3, O_2\rangle$ *and the function definitions* $f_1 = x_1, f_2 = x_2, f_3 = x_3$ *has no planar acyclic realisation.* □

Theorem 4.3 *The main construction produces a planar acyclic circuit whenever one exists.*

Proof. The result follows from Theorems 4.1 and 4.2. □

5 Conclusion

Our results give a complete solution to the problem of determining whether or not a given I/O specification can be realised as a planar acyclic circuit constructed from two-input (1-directional) gates. The extension to k-input d-directional gates is straightforward for $d \leq 2$ and any $k \geq 2$. However, if we allow 3-directional gates then specifications such as that in Corollary 4.1 become realisable.

It is well known that any graph can be embedded in three-dimensional Euclidean space in such a way that each edge corresponds to a straight line segment in R^3. For a proof of this result, see, for example, [Wh, Chapter 6]. Let $2LG$ be the two layer grid, i.e., the subset of points $\langle x, y, z\rangle \in R^3$ where $0 \leq z \leq 1$ and at least two of x, y, z are integers. If we allow each edge to be constructed from a set of straight line segments then a simple variant of our construction can be used to show that any I/O specification can be realised as an acyclic circuit embedded in $2LG$ with no two edges meeting at an interior point.

If we restrict our attention to 1-directional specifications then Theorem 2.1 shows that any such specification can be realised as a planar acyclic circuit. The layout method used in the proof of this result may in some cases produce a circuit which is much larger than is necessary. As a challenge to planar acyclic circuit designers, we offer the following

Open Problem. *Let S be a 1-directional I/O specification for which there is an acyclic circuit which can be embedded in the plane with g gates and c crossings. Is there always a planar acyclic circuit for such an S which is of size $O(g+c)$ and has no crossings?*

References

[A] A. Aggarwal, *On simulation and interrelations between Boolean and VLSI circuits*, Proc. 21st Annual Allerton Conf. (1983), 258-265.

[BM] A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier (1975).

[vzG1] J. von zur Gathen, *Parallel arithmetic computations : A survey*, Proc. Mathematical Foundations of Computer Science 1986, LNCS Vol. 233, Springer-Verlag (1986), 93-112.

[vzG2] J. von zur Gathen, *Feasible Arithmetic Computations : Valiant's Hypothesis*, Journal of Symbolic Computation **4** (1987), 137-172.

[GH] A. K. Gupta and S. E. Hambrusch, *Optimal three-dimensional layouts of complete binary trees*, Information Processing Letters **26** (1987), 99-104.

[LR] F. T. Leighton and A. L. Rosenberg, *Three-dimensional circuit layouts*, SIAM Journal on Computing **15** (1986), 793-813.

[L] T. Lengauer, *The communication complexity of VLSI circuits*, Proc. AMS Symp. in Applied Mathematics **31** (1985), 59-82.

[LT] R. J. Lipton and R. E. Tarjan. *Applications of a planar separator theorem*, SIAM Journal on Computing **9** (1980), 615-627.

[McC1] W. F. McColl, *Planar crossovers*, IEEE Transactions on Computers C-30 (1981), 223-225.

[McC2] W. F. McColl, *On the planar monotone computation of threshold functions*, Proc. 2nd Annual Symp. on Theoretical Aspects of Computer Science. LNCS Vol. 182, Springer-Verlag (1985), 219-230.

[MRK] G. L. Miller, V. Ramachandran and E. Kaltofen, *Efficient parallel evaluation of straight-line code and arithmetic circuits*, VLSI Algorithms and Architectures. LNCS Vol. 227, Springer-Verlag (1986), 236-245.

[P] F. P. Preparata, *Optimal three-dimensional VLSI layouts*, Mathematical Systems Theory **16** (1983), 1-8.

[Re] J. Reif, *Logarithmic depth circuits for algebraic functions*, Proc. 24th IEEE Symp. on Foundations of Computer Science (1983), 138-145.

[Ro] A. L. Rosenberg, *Three-dimensional VLSI : A case study*, Journal of the ACM **30** (1983), 397-416.

[S1] J. E. Savage, *Planar circuit complexity and the performance of VLSI algorithms*, VLSI Systems and Computations. H. T. Kung, B. Sproull and G. Steele (eds.), Computer Science Press (1981), 61-68. Expanded version appears as INRIA Report No.77, INRIA, Rocquencourt, France (1981).

[S2] J. E. Savage, *The performance of multilective VLSI algorithms*, Journal of Computer and System Sciences **29** (1984), 243-273.

[U] J. D. Ullman, *Computational Aspects of VLSI*, Computer Science Press (1984).

[We] I. Wegener, *The Complexity of Boolean Functions*. Wiley-Teubner (1987).

[Wh] A. T. White, *Graphs, Groups and Surfaces*. North-Holland (1984).

[Y1] M. Yannakakis, *Four pages are necessary and sufficient for planar graphs*, Proc. 18th ACM Symp. on Theory of Computing (1986), 104-108.

[Y2] M. Yannakakis, *Linear and book embeddings of graphs*, VLSI Algorithms and Architectures. LNCS Vol. 227, Springer-Verlag (1986), 226-235.